

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-012118
 (43)Date of publication of application : 22.01.1993

(51)Int.Cl. G06F 12/08

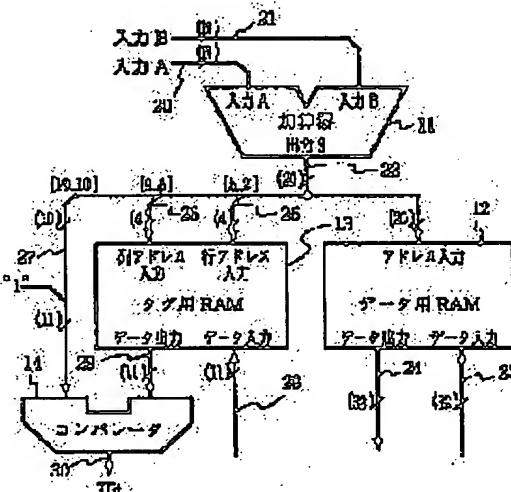
(21)Application number : 03-164862 (71)Applicant : NEC CORP
 (22)Date of filing : 05.07.1991 (72)Inventor : NAKAYAMA TAKASHI

(54) CACHE MEMORY DEVICE WITH ADDRESS ADDER

(57)Abstract:

PURPOSE: To start an access to a memory from the part where the output of an address adder is defined and to shorten read-out time by arranging the row address of the memory at a low-order digit than the column address the memory.

CONSTITUTION: The address input signal of a tag RAM 13 is divided into a 4-bit row address signal 25 and a 4-bit column address signal 26. For calculating the address, two input binary numbers are added by an adder 11, the low-order digit output of the adder 11 is stored in a data memory 12 as the address, and in which address of the data memory 12 the data were stored is stored in the tag memory 13. A comparator 14 compares the high-order digit output of the adder 11 with the output of the tag memory 13, inputs the low-order digit part of the low-order digit of the output of the adder 11 to the row address of the tag memory 13 and inputs the high-order digit part of the low-order digit of the output of the adder 11 to the column address of the tag memory 13. Thus, the memory capable of eliminating the wasteful time in delay time can be obtained.



LEGAL STATUS

[Date of request for examination] 14.11.1997

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3289284

[Date of registration] 22.03.2002

[Number of appeal against examiner's

decision of rejection]

[Date of requesting appeal against
examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平5-12118

(43)公開日 平成5年(1993)1月22日

(51)Int.Cl.⁵

G 0 6 F 12/08

識別記号

府内整理番号

3 1 0 Z 7232-5B

F I

技術表示箇所

審査請求 未請求 請求項の数1(全9頁)

(21)出願番号 特願平3-164862

(22)出願日 平成3年(1991)7月5日

(71)出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72)発明者 中山 貴司

東京都港区芝五丁目7番1号日本電気株式
会社内

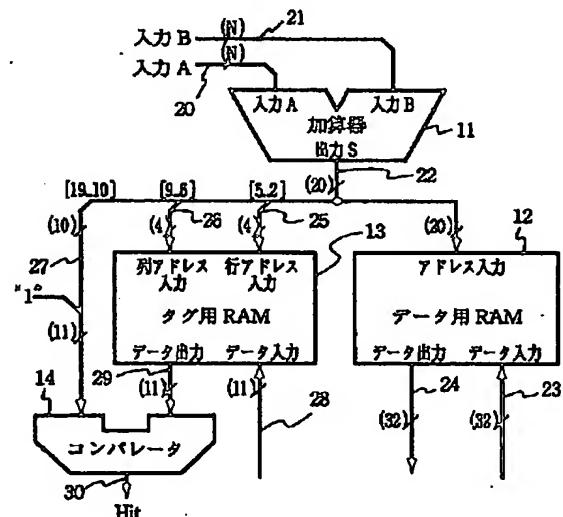
(74)代理人 弁理士 内原 晋

(54)【発明の名称】 アドレス加算器付きキヤツシユメモリ装置

(57)【要約】

【構成】アドレス計算のために加算器11により2つの入力2進数を加算し、この加算器11の下位桁出力をアドレスとしてデータ用メモリ12にデータを格納し、このデータ用メモリ12がどのアドレスのものかをタグ用メモリ13に記憶する。比較器14は加算器11の上位桁出力とタグ用メモリ13の出力を比較し、加算器11の出力の下位桁の下位桁部分をタグ用メモリ13の行アドレスに入力し、加算器11の出力の下位桁の上位桁部分をタグ用メモリ13の列アドレスに入力する。

【効果】タグ用メモリ13の行アドレスによるアクセス時間が列アドレスによるアクセス時間より短いので、これに加算器11の演算時間を合わせて、遅延時間に無駄のないメモリを得る。



1

【特許請求の範囲】

【請求項1】 アドレス計算に2つの入力2進数を加算する加算器と、この加算器の下位桁出力をアドレスとしてデータを格納するデータ用メモリと、このデータ用メモリがどのアドレスのものかを記憶するタグ用メモリと、前記加算器の上位桁出力と前記タグ用メモリの出力を比較する比較器とを備え、前記加算器の出力の下位桁の下位桁部分を前記タグ用メモリの行アドレスに入力し、前記加算器出力の下位桁の上位桁部分を前記タグ用メモリの列アドレスに入力することを特徴とするアドレス加算器付きキャッシュメモリ装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、計算機の記憶装置に使われ半導体メモリを用いたアドレス加算器つきキャッシュメモリ装置に関する。

【0002】

【従来の技術】 マイクロプロセッサでは、メモリをアクセスするときに、次のようなベース・アドレッシング・モードでアクセスするのが一般的である。

【0003】

命令 `load disp[r0], r2`
動作 `memory(r0+disp) → r2`

この例では、レジスタ r_0 の内容とディスプレースメント値 ($disp$) の和をアドレスとしてメモリを参照し、そのメモリの内容をレジスタ r_2 に加算する。その他のアドレッシング・モードでもアドレス演算に加算を伴うものが多い。実際のプログラムでは、アドレス計算に加算を行なう場合がほとんどである。

【0004】 また、最近の RISC 型プロセッサでは、プロセッサの処理が高速化し、主記憶の速度がそれに追い付かないため、小容量だが高速なキャッシュメモリを付加するのが一般的である。そのため、アドレス計算を行なってキャッシュメモリを読み出すまでの時間が、プロセッサの速度を決定する主な原因となっている。そのため、プロセッサの演算速度を向上させるには、加算処理とキャッシュメモリへのアクセスを高速化することが求められている。

【0005】 図5は従来例のアドレス加算器付きキャッシュメモリ装置のブロック図を示す。この例では、アドレス $S = A + B$ に位置するメモリをアクセスでき、キャッシュメモリとしては、最も高速でかつ回路が単純なダイレクト・マップ方式を用いている。20ビットの入力信号 (A) 20および20ビットの入力信号 (B) 21は、20ビットの桁上げ伝播加算器 11 により加算されて20ビットの出力信号 (S) 22を得る。この出力信号 22 をアドレスとして32ビット $\times 2^{20}$ 語のメモリ (RAM) 12 に入力し、また32ビットの入力データ信号 23 および32ビットの出力データ信号 24 と接続される。11ビット $\times 2^8$ 語のメモリ (RAM) 13 は

10

20

30

40

50

2

信号 (S) 22 の下位桁 8 ビット (ビット 9~2) 信号 31 をメモリのアドレス入力とし、11ビットの入力データ信号 28 および11ビットの出力データ信号 29 と接続される。

【0006】 信号 26 は信号 S (22) の上位桁 10 ビット (ビット 19~10) に1ビットの「1」を付加した 11 ビット信号で、出力データ信号 29 とコンパレータ 14 で比較され一致を示すヒット出力信号 30 を出力する。

【0007】 データ用メモリ 12 に格納されるデータは、4語毎のブロックで管理され、各ブロック毎にタグが付加されて、タグ用メモリ 13 に格納されている。このタグは、各ブロックがアドレス空間のどこに位置するかのタグ情報 (アドレスの上位桁) と、各ブロックに有効なデータが格納されているかを示すフラグ (バリッドフラグ) から構成される。コンパレータ 14 によって、アドレスの上位桁 26 とタグ用メモリ 13 のタグ情報とが一致し、かつ、バリッドフラグが有効であることが検出されると、データ用メモリ 12 の内容が有効となる。

20 この状態をキャッシュが「ヒット」したと呼び、逆の状態を「ミス」したと呼ぶ。キャッシュがヒットすればプロセッサは処理を続行し、ミスした場合は対応するブロックのデータを改めて主記憶からデータ用メモリ 12 に読み込み、対応するタグ情報をタグ用メモリ 13 に書き込む。

【0008】 入力 A (20) と入力 B (21) の値が与えられると加算器 11 で加算 ($S = A + B$) が行われ、時間 T_a 後にアドレス 22 が確定する。アドレス 22 が確定してからアクセス時間 T_r 後に出力データ 29 が確定する。このデータ 29 の確定した時間 T_c 後にヒット信号 30 が確定する。

【0009】 入力 A (20) と入力 B (21) の値が与えられてから、ヒット信号 (30) が確定するまでの時間 T_o が、このアドレス加算器付きキャッシュメモリの性能を示す。

$$T_o = T_a + T_r + T_c \quad \dots (1)$$

ここで、 $T_a = 50 \text{ ns}$, $T_r = 30 \text{ ns}$, $T_c = 10 \text{ ns}$, ならば $T_o = 90 \text{ ns}$ となる。

【0010】 (1) 式は、一見正しいように見えるが、加算器 11 とメモリ 13 の詳細な構成を考慮すると異なることがわかる。

【0011】 図6は加算器 11 の詳細な構造を示すブロック図である。加算器 11 は 20 桁のリップル桁上げ方式の桁上げ伝播加算器で、20個の全加算器 40~59 から構成される。これら全加算器は、入力 A, B, C_{in} の1の数をかぞえて 2 桁の2進数 (C_{out}, S) を出力する。全加算器の出力 C_{out} は上の桁の全加算器の入力 C_{in} に接続されている。そのため、入力 A (20), B (21) が与えられてから、加算結果 S (22) の第 k ビットが確定する時間 $T_a (k)$ は、全加算

器の遅延時間を T_f とすると、

$$T_a(k) = (k+1) \times T_f \quad \dots (2)$$

で与えられる。つまり、LSB (Least Significant Bit) 付近の結果は早く得られるが、MSB (Most Significant Bit) 付近の結果が確定するのは時間がかかる。

【0012】同様の考察をメモリに対して行なってみる。図7はスタティック・ランダムアクセス・メモリ (SRAM) の一般的な構造を示すブロック図である。Nビット ($N=8$) のアドレス入力信号31は、nビット ($m=4$) の行アドレス信号25とmビット ($n=4$) の列アドレス信号とに分けられ、行デコーダ71が $n=4$ ビットのアドレス入力信号25をデコードして2ⁿ本のワードライン信号72を生成する。2ⁿ本あるメモリセル73はそれぞれ1ビットの情報を記憶し、2本1組で2ⁿ組あるビットライン74の1組はメモリセル73にデータを読み書きする真偽1組の信号となる。ビットライン74のマルチプレクサ75は20チャネルのMOSトランジスタで構成され、パッファ76は1ビット分の入力データ28をマルチプレクサ75に伝える。読み出し用センスアンプ77はマルチプレクサ75の出力を差動入力で増幅し、列デコーダ78はmビットの入力*

$$T_r(R) = T_{rd} + T_{ce} + T_{se} + T_{ap} \quad \dots (3)$$

$$T_r(C) = T_{cd} + T_{se} + T_{ap} \quad \dots (4)$$

ここで、 T_{rd} を行デコーダ71の遅延時間、 T_{cd} を列デコーダ78の遅延時間、 T_{ce} をメモリセル73のアクセス時間、 T_{se} をセレクタ75の遅延時間、 T_{ap} をセンスアンプ77の遅延時間とする。また T_{rd} は T_{cd} とほぼ等しいので、 $T_r(R)$ は $T_r(C)$ よりも、約 T_{ce} 分だけ時間がかかることになる。

【0016】ここで具体例として、アドレス8ビットの加算器付きメモリを考える。 $N=8$ 、 $n=4$ 、 $m=4$ であり、行アドレス25がアドレス31の上位4ビット、列アドレス26がアドレス31の下位4ビットであるものとする。

【0017】図8はこの場合の各部の遅延時間の特性図を示す。横軸はアドレスのビット番号、縦軸は時刻である。各部の遅延時間は、次のとおりとする。

$$T_f = 2.5 \text{ ns}, T_{rd} = 7.5 \text{ ns}, T_{cd} = 7.5 \text{ ns}, T_{ce} = 12.5 \text{ ns}, T_{se} = 3.0 \text{ ns}, T_{ap} = 7.0 \text{ ns} \quad \dots (5)$$

図5の例ではメモリ13のアドレスとして用いられるのは、加算器22の下位桁のみであり、上位桁加算を待たずに済む。(1)式は次の(6)式で置き換えられる。

$$T_o = T_a(9) + T_r + T_c \quad \dots (6)$$

図8では、 $T_a(9) = 25 \text{ ns}$ 、 $T_r = 30 \text{ ns}$ 、 $T_c = 10 \text{ ns}$ 、 $T_o = 65 \text{ ns}$ であり、(1)式による 90 ns より良い性能が出ていることが分かる。しかし、遅延のネックとなっているのはアドレス31のMSBであり、列アドレスは早く得られるが、行アドレス側

*アドレス信号26をデコードしてマルチプレクサ75を制御する。

【0013】 $2^2 \times 0$ 語のメモリセルは、 2^0 行 $\times 2^0$ 列の平面上に配置されている。ここで、 $N=n+m$ とする。20ビットのアドレスがnビットの行アドレス25とmビットの列アドレス26に分割される。

【0014】メモリの読み出しは、以下の手順で行なわれる。アドレス31が与えられると、行デコーダ71で行アドレス25をデコードし、ワードラインのひとつ(72)が「1」になる。すると、その行の2ⁿ個のメモリセルが選択され、それぞれのビットラインにメモリセルの値(一方にはメモリセルの値、もう一方にはその反転値)を出力する。その間に、列デコーダ78で列アドレス26をデコードし、セレクタ75を制御してビットライン74を選択しておく。すると、選択したビットライン(74)上の値がセンスアンプ77に伝えられ、メモリセル(73)の値が出力データ29として出力される。

【0015】メモリの読み出しがアクセス時間は、行アドレスからのアクセス時間 $T_r(R)$ と、列アドレスからのアクセス時間 $T_r(C)$ の2通り規定できる。

$$T_r(R) = T_{rd} + T_{ce} + T_{se} + T_{ap} \quad \dots (3)$$

$$T_r(C) = T_{cd} + T_{se} + T_{ap} \quad \dots (4)$$

が遅いためメモリの読み出しが遅れている。ハードウェアが有効に使われていないことが分かる。

【0018】

【発明が解決しようとする課題】上述した従来例のアドレス演算器付きキャッシュメモリ装置では、読み出し時間が長いという欠点がある。これは、メモリの読み出し時間は、行アドレスからのアクセス時間 $T_r(R)$ よりも、列アドレスからのアクセス時間 $T_r(C)$ の方が短いことを有効に活用できていないためである。この有効活用のためには、行アドレスをなるべく早くメモリに伝えるように加算器の構成を変える必要がある。

【0019】本発明の目的は、メモリの行アドレスを列アドレスより下位桁にすることによって、アドレス加算器の出力が確定した部分からメモリのアクセスを開始させ、読み出し時間を短くしたキャッシュメモリ装置を提供することにある。

【0020】

【課題を解決するための手段】本発明のアドレス加算器付きキャッシュメモリ装置の構成は、アドレス計算に2つの入力2進数を加算する加算器と、この加算器の下位桁出力をアドレスとしてデータを格納するデータ用メモリと、このデータ用メモリがどのアドレスのものかを記憶するタグ用メモリと、前記加算器の上位桁出力と前記タグ用メモリの出力を比較する比較器とを備え、前記加算器の出力の下位桁の下位桁部分を前記タグ用メモリの行アドレスに入力し、前記加算器出力の下位桁の上位桁

部分を前記タグ用メモリの列アドレスに入力することを特徴とする。

【0021】

【実施例】図1は本発明の一実施例のアドレス加算器付きキャッシュメモリのブロック図である。この実施例では、アドレス $S = A + B$ に位置するメモリをアクセスできる。キャッシュメモリは、最も高速でかつ回路が単純なダイレクト・マップ方式を用いている。本実施例が図5の実施例と相違している点は、タグ用RAM13のアドレス入力信号31が、4ビットの行アドレス信号25および4ビットの列アドレス信号26に分けられていることである。

【0022】加算器11は、図6で示したものと同じ構成であり、遅延時間 T_a には(2)式が成立つ。メモリ12は図7で示したものと同じ構成であり、遅延時間 T_r には(3), (4)式が成立つ。本実施例が従来例(図2)と異なるのは、メモリ13へのアドレスが、加算器出力22の下位桁側を行アドレスしたことである。

【0023】入力A(20)と入力B(21)の値が与えられると加算器11で加算($S = A + B$)が行なわれ、(2)式より時間 T_a (5)後に行アドレスが、時間 T_a (9)後に列アドレスが確定する。行アドレスが確定してからアクセス時間 T_r (R)後の時刻と、列アドレスが確定してからアクセス時間 T_r (C)後の時刻の、遅い方の時刻にメモリ出力が確定する。また、メモリ出力115と加算器出力113のうち遅い方から T_c で構成される。

【0024】全体遅延時間は、

$$T_o = \max (T_a (5) + T_r (R), T_a (9) + T_r (C), T_a (19) + T_c) \dots (7)$$

この式に(2), (3), (4), (5)式を代入するところのようになる。

$$T_o = \max (15 + 30, 25 + 17, 5, 50) + 10 = 60 \text{ ns}$$

図1の各部の遅延時間を図2に示す。(7)式の各変数の値は(5)式(図8で使用したもの)と同じである。

図2では各部分の遅延時間は図8と同じであるが、図8*

$$T_o = \max (10 + 30, 25 + 17, 5, 40) + 10 = 50 \text{ ns}$$

であり、図2の場合より 10 ns 速くなっている。これは行アドレスの生成とコンパレータへの入力を更に速くしたためである。列アドレスの生成は高速化していないが、これによる性能向上はこの例ではないためである。

【0031】本実施例のアドレス加算器は、桁上げ選択加算器(Carry Select Adder)を使用したが、桁上げ先見機構を用いた加算器(Carry Look-Ahead Adder)でもよく、また、本実施例ではメモリとしてスタティック・メモリ(SRAM)を用いたが、ダイナミック・メモリ(DRAM)であっても、読み出し専用メモリ(ROM)であってもかまわない。

*より 5 ns 速くなっている。これは行アドレスに速い信号を割当てたためである。

【0025】図3は本発明の第2の実施例として他の加算器を用いた場合の加算器のブロック図である。

【0026】図において、加算器11は、ビット3~5とビット17~19において桁上げ選択加算器41, 42, 44, 45に、それ以外のビットはリップル桁上げ加算器40, 43になっている。

【0027】リップル桁上げ加算器41~45は、入力A, B, Cinの1の数をかぞえて2桁の2進数(Cout, S)を出力する全加算器が3個(43は1個)で構成される。

【0028】リップル桁上げ加算器40の桁上げ出力信号(Cout)49によって3ビットのセレクタ46がリップル桁上げ加算器41, 42の和出力信号(S)を切換える。ゲート48はリップル桁上げ加算器40~42の桁上げ出力信号(Cout)からリップル桁上げ加算器43への桁上げ入力信号(Cin)を生成し、リップル桁上げ加算器43の桁上げ出力信号(Cout)50によって3ビットのセレクタ47がリップル桁上げ加算器44, 45の和出力信号(S)を切換える。

【0029】リップル桁上げ加算器41, 42とセレクタ46および、リップル桁上げ加算器44, 45とセレクタ47とは、3ビットの桁上げ選択加算器41, 44は桁上げ信号49, 50が1の場合の3ビットの加算を、全加算器42, 45は桁上げ信号49, 50が0の場合の3ビットの加算を行なっており、桁上げ信号49, 50が確定した時点でセレクタ46, 47で加算器41または42(44または45)の和出力を選択する。同時に、ゲート48によって全加算器43への桁上げ入力信号が確定できる。

【0030】図3の加算器を使用した場合の図1の各部の遅延時間を図4に示す。(7)式の各変数の値は(5)式(図8で使用したもの)と同じである。また、セレクタ46とゲート48の遅延時間は共に 2.5 ns とする。図4では、

$$5, 40) + 10 = 50 \text{ ns}$$

【0032】

【発明の効果】以上説明したように、本発明によるアドレス加算器付きメモリは、次の2つの効果を有する。

【0033】(1) 演算時間が短い。

従来例の 65 ns に対し、実施例1(図6)では 60 ns 、実施例2では 50 ns と早くなっている。メモリの行アドレスによるアクセス時間が列アドレスによるアクセス時間より短いことを利用し、加算器の演算時間の分布をこれに合せることによって、遅延時間に無駄の無いメモリが得られる。

【0034】(2) ハードウェア量はほとんど増加しない。

実施例1ではハードウェア量は従来例と同じであり、実施例2では従来例より全加算器6個、3ビットのセレクタが2個、ゲート1個だけ増加しただけで済んでいる。

【図面の簡単な説明】

【図1】本発明の一実施例のハードウェア構成を示すブロック図。

【図2】図1の実施例の効果を示す遅延時間の模式図。

【図3】本発明の第2の実施例で使用した加算器のブロック図。

【図4】第2の実施例の効果を示す遅延時間の模式図。

【図5】従来例のキャッシュメモリ装置のハードウェア構成を示すブロック図。

【図6】図5で使用した加算器の構造を示すブロック図。

【図7】従来例で使用したメモリの構造を示すブロック図。

【図8】従来例の演算時間を示す遅延時間の模式図。

【符号の説明】

1 1 柄上げ伝播加算器

1 2 データ用メモリ

1 3 タグ用メモリ

1 4 コンパレータ

2 0, 2 1 入力信号

2 2 出力信号S

2 3, 2 8 入力データ信号

2 4, 2 9 出力データ信号

2 5, 2 6, 3 1 アドレス信号

3 0 ビット出力信号

4 0 ~ 4 5 リップル桁上げ加算器

4 6, 4 7 セレクタ

10 4 8 ゲート

4 9, 5 0 柄上げ信号

5 1 ~ 7 0 全加算器

7 1 行デコーダ

7 2 ワードライン信号

7 3 メモリセル

7 4 ビットライン

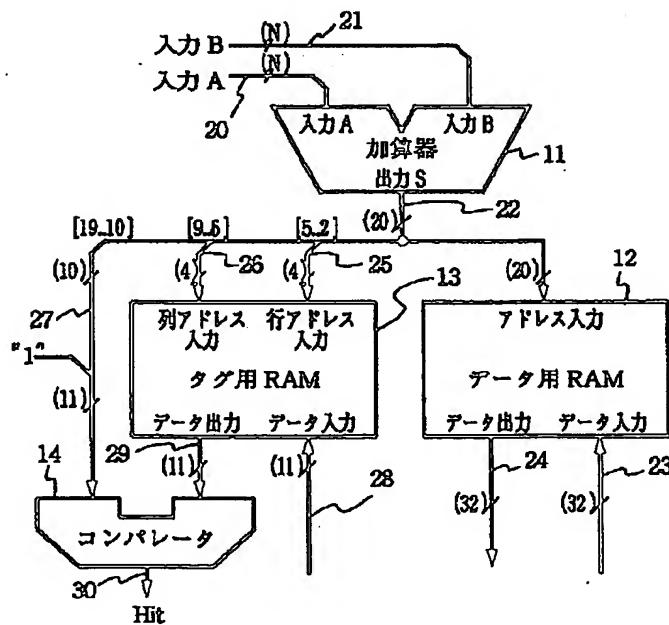
7 5 マルチブレクサ

7 6 パッファ

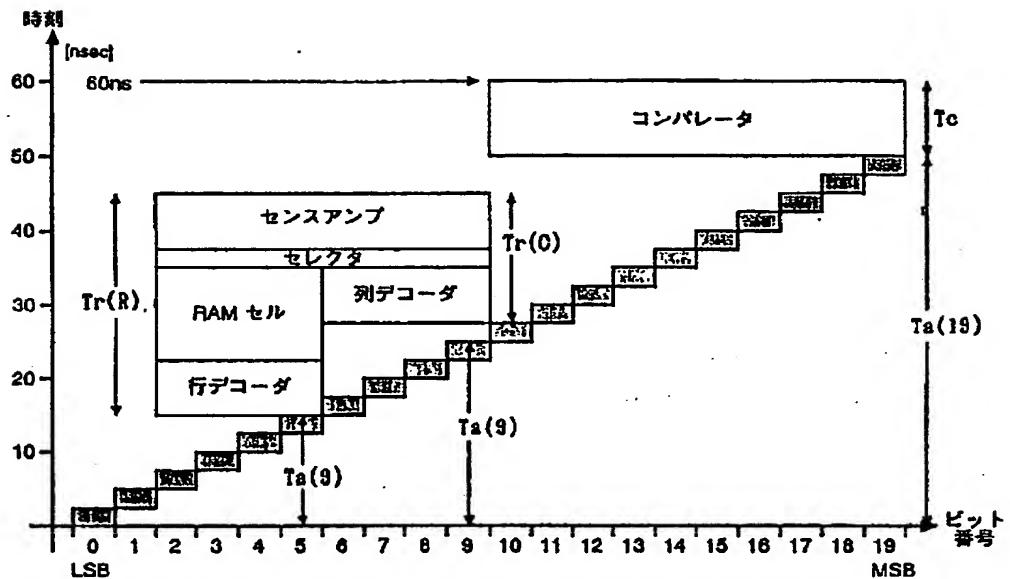
7 7 センスアンプ

20 7 8 列デコーダ

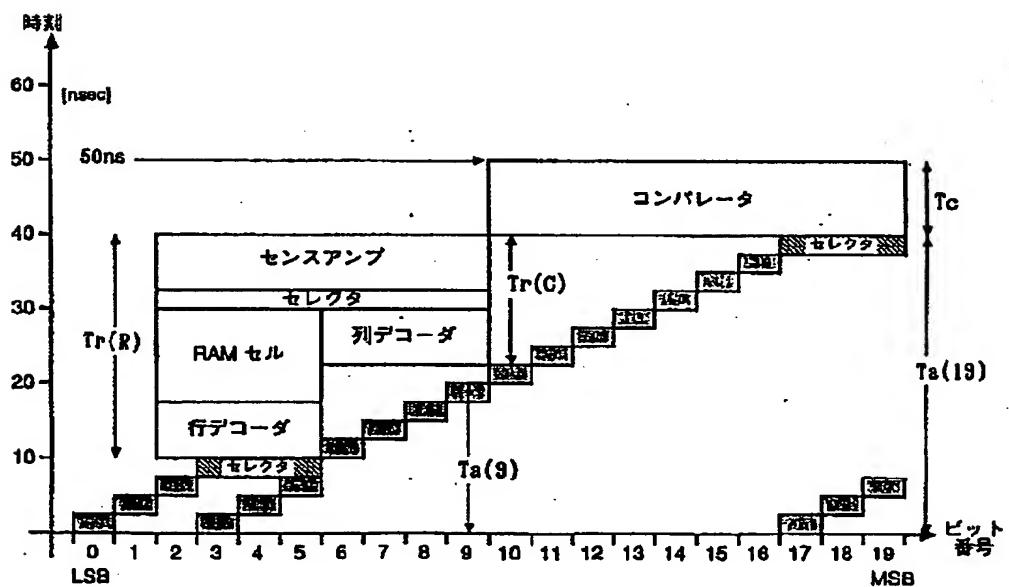
【図1】



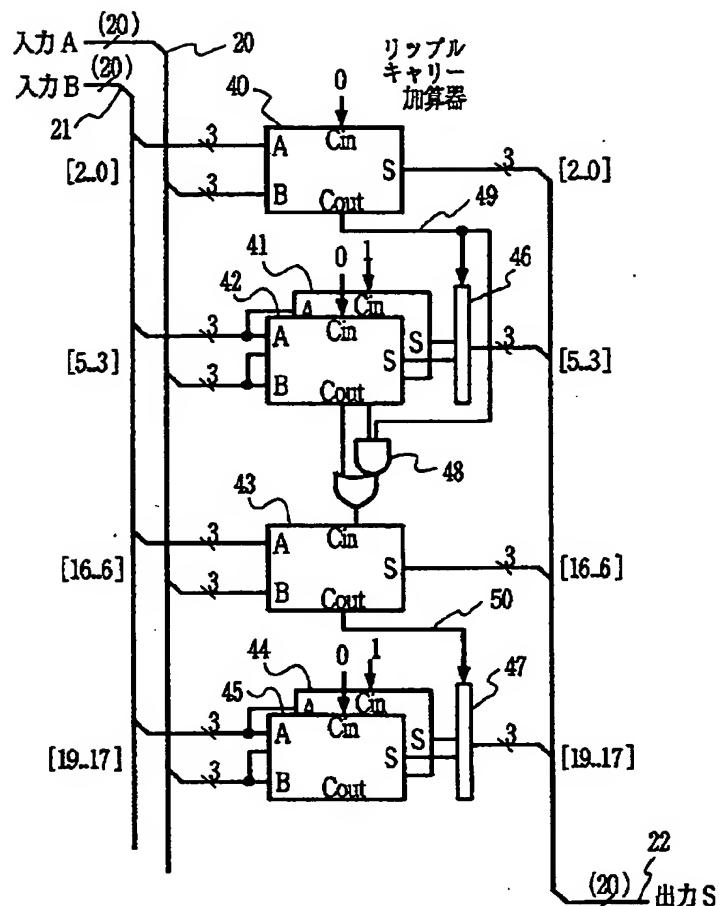
【図2】



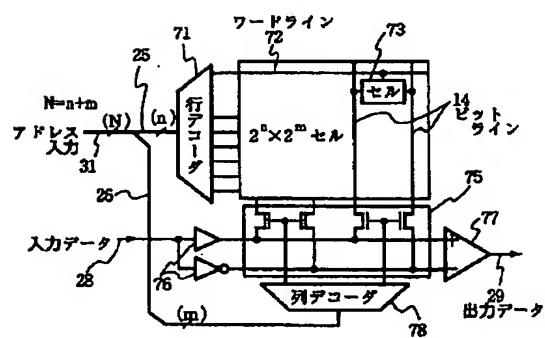
【図4】



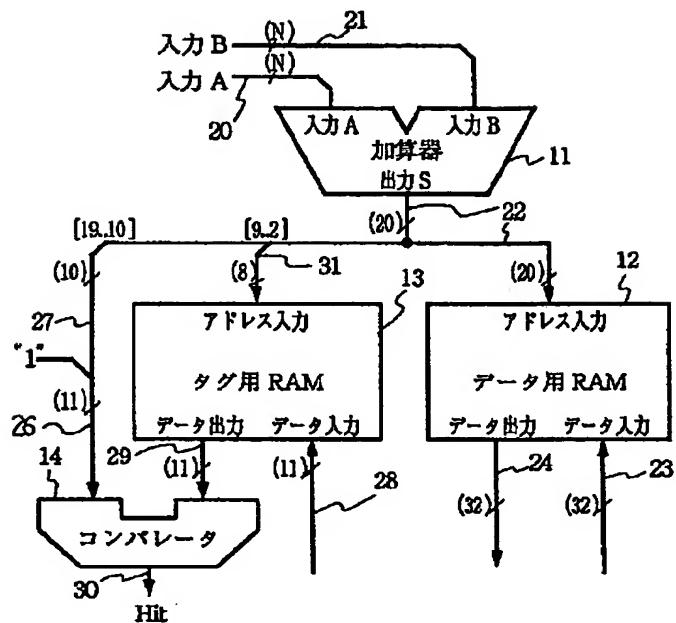
【図3】



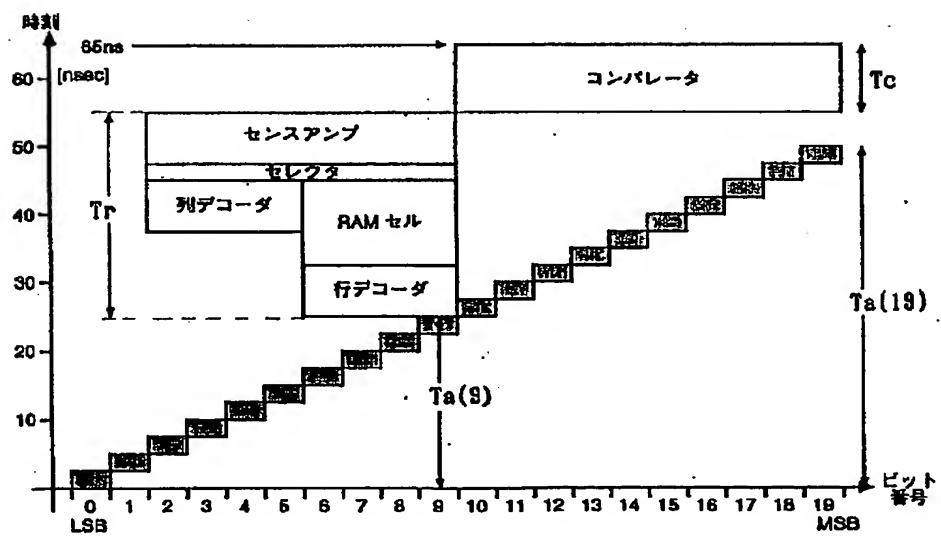
【図7】



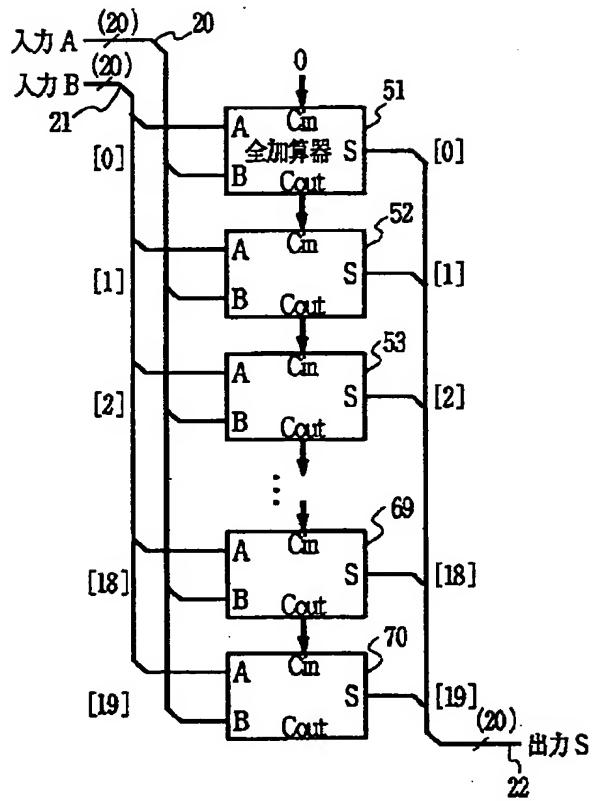
[5]



[图8]



【図6】



THIS PAGE BLANK (USPS)